eip ESCUELA INTERNACIONAL POSGRADOS

Máster en Programación avanzada en Python para Hacking, BigData y Machine Learning

Fundamentos de IA y Machine Learning

LECCIÓN 04

Modelos de Regresión y Clasificación II

ÍNDICE

INTRODUCCIÓN

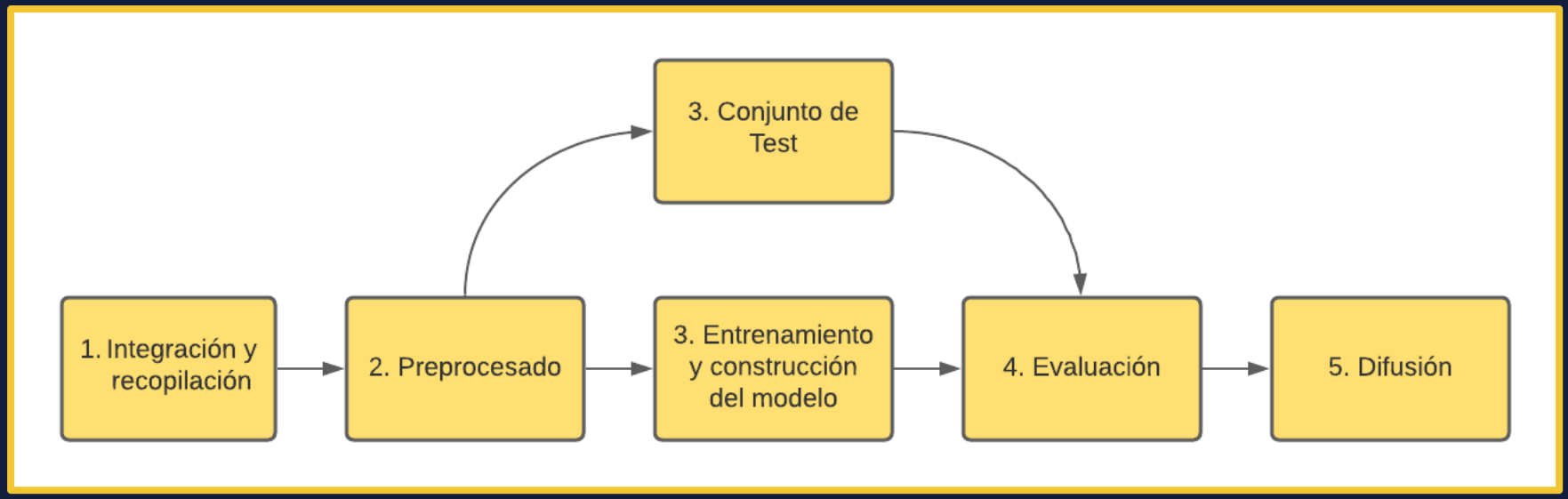
Una vez que conocemos las técnicas de entrenamiento y evaluación, es el momento de exponer distintos modelos de aprendizaje automático y los algoritmos utilizados para su entrenamiento.

OBJETIVOS

Al finalizar esta lección serás capaz de:

1. Generar, entrenar e interpretar una red neuronal artificial.
2. Conocer el fundamento de los vectores de máquina soporte para clasificación (SVM) y regresión (SVR).
3. Hacer uso de ensembles conociendo sus fundamentos.

1.1. Contextualización de la lección



1.2. Formulación de un problema de regresión

Cualquier problema de regresión se puede formular de la siguiente forma:

- Disponemos de un espacio de entrada $\boldsymbol{X}$ compuesto por patrones etiquetados con $Y \subseteq \mathbb{R}$.
- Cada patrón se representa por un vector de características de dimensión K, $\boldsymbol{x} \in \boldsymbol{X} \subseteq \mathbb{R}^K$ y un valor continuo $y \in Y \subseteq \mathbb{R}$.
- El objetivo es aprender una función f que relacione los datos del espacio de entrada $\boldsymbol{X}$ al conjunto de valores continuos finito Y.
- El conjunto de entrenamiento $\boldsymbol{T}$ está compuesto de N patrones:

$$T = (\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{y_i} \in Y (i = 1, \dots, n), \text{ con } x_i = \left(x_{i,1}, x_{i,2}, \dots, x_{i,K}\right).$$

1.3. Formulación de un problema de clasificación

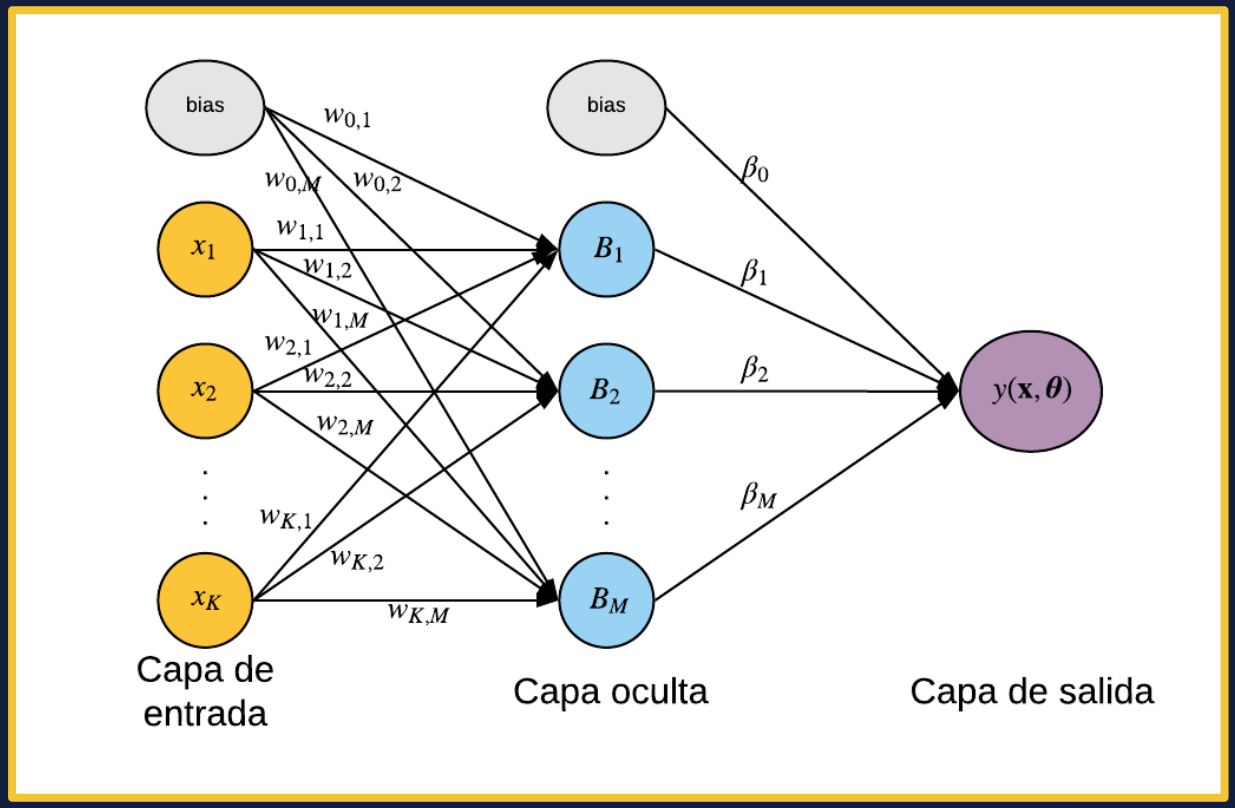
Cualquier problema de clasificación se puede formular de la siguiente forma:

- Disponemos de un espacio de entrada $\boldsymbol{X}$ compuesto por patrones etiquetados con $C = \{C_1, C_2, \dots, C_Q\}$ donde Q es el número de clases.
- Cada patrón se representa por un vector de características de dimensión K, $\boldsymbol{x} \in \boldsymbol{X} \subseteq \mathbb{R}^K$ y una etiqueta de clase $y \in C$.
- El objetivo es aprender una función f que relacione los datos del espacio de entrada $\boldsymbol{X}$ al conjunto finito C.
- El conjunto de entrenamiento $\boldsymbol{T}$ está compuesto de N patrones:

$$T = (\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{y_i} \in C (i = 1, \dots, n), \text{ con } x_i = \left(x_{i,1}, x_{i,2}, \dots, x_{i,K}\right).$$

2. Redes neuronales artificiales – Fundamentos y modelo

$$y(x, \boldsymbol{\theta}) = \beta_0 + \sum_{j=1}^{M} \beta_j \, B_j(x, w_j)$$



2. Redes neuronales artificiales – Fundamentos y modelo

Funciones de activación

- **Modelo aditivo:**

$$B_j(\mathbf{x}, \boldsymbol{w_j}) = h(w_{0,j} + w_{1,j}x_1 + w_{2,j}x_2 + \cdots + w_{K,j}x_K)$$

- **Modelo multiplicativo:**

$$B_j(x, w_j) = x_1^{w_{1,j}} \cdot x_2^{w_{2,j}} \cdot \ldots \cdot x_K^{w_{K,j}}$$

2. Redes neuronales artificiales – Fundamentos y modelo

Considerando el espacio característico de entrada de las funciones base, podemos clasificarlas en

- **Funciones locales o *kernels*:** presentan un mayor valor sobre una región específica del espacio de entrada. Son buenas para aproximar datos aislados, pero más pobres en entornos globales y cuando el número de entrada es elevado. Un ejemplo son las funciones de base radial o RBF.

- **Funciones globales o de proyección:** son mejores en el caso contrario, es decir, se comportan mejor para entornos globales, pero de forma peor en la aproximación de datos aislados. Un ejemplo son las unidades sigmoide (SU) y las unidades producto (PU).

2. Redes neuronales artificiales – Fundamentos y modelo

Redes neuronales sigmoide

$$B_j(\mathbf{x}, \mathbf{w_j}) = \frac{1}{1 + e^{-(w_{0,j} + w_{1,j}x_1 + w_{2,j}x_2 + \cdots + w_{K,j}x_K)}} = \frac{1}{1 + e^{-(w_{0,j} + \sum_{i=1}^{K} w_{i,j}x_i)}}$$

2. Redes neuronales artificiales – Fundamentos y modelo

Redes neuronales producto

$$B_j(\mathbf{x}, \mathbf{w_j}) = x_1^{w_{1,j}} \cdot x_2^{w_{2,j}} \cdot \ldots \cdot x_K^{w_{K,j}} = \prod_{i=1}^{K} x_i^{w_{i,j}}$$

2. Redes neuronales artificiales – Entrenamiento

Algoritmo de retropropagación

1. Se propaga la entrada hacia la salida, obtiene el valor estimado por nuestra función:

$$\hat{y}(\boldsymbol{x}, \boldsymbol{\theta}) = \beta_0 + \sum_{j=1}^{M} \beta_j \, B_j(\boldsymbol{x}, \boldsymbol{w_j})$$

2. Obtenemos el error cometido por la red $E = \left(y - \hat{y}(\boldsymbol{x}, \boldsymbol{\theta})\right)^2$.
3. Se calcula el gradiente del error:

$$\nabla E = \frac{\partial E}{\partial \boldsymbol{\theta}} = \left\{ \frac{\partial E}{\partial w_{0,1}}, \ldots, \frac{\partial E}{\partial w_{0,M}}, \ldots, \frac{\partial E}{\partial w_{N,M}}, \frac{\partial E}{\partial \beta_0}, \ldots, \frac{\partial E}{\partial \beta_M} \right\}$$

4. Una vez que hemos obtenido el gradiente, debemos actualizar los pesos. Siendo $\Delta w_{i,j} = \frac{\partial E}{\partial w_{i,j}}$, se tiene la siguiente expresión:

$$w_{i,j} = w_{i,j} - \eta \Delta w_{i,j} - \mu \left(\eta \Delta w_{i,j}(t-1) \right)$$

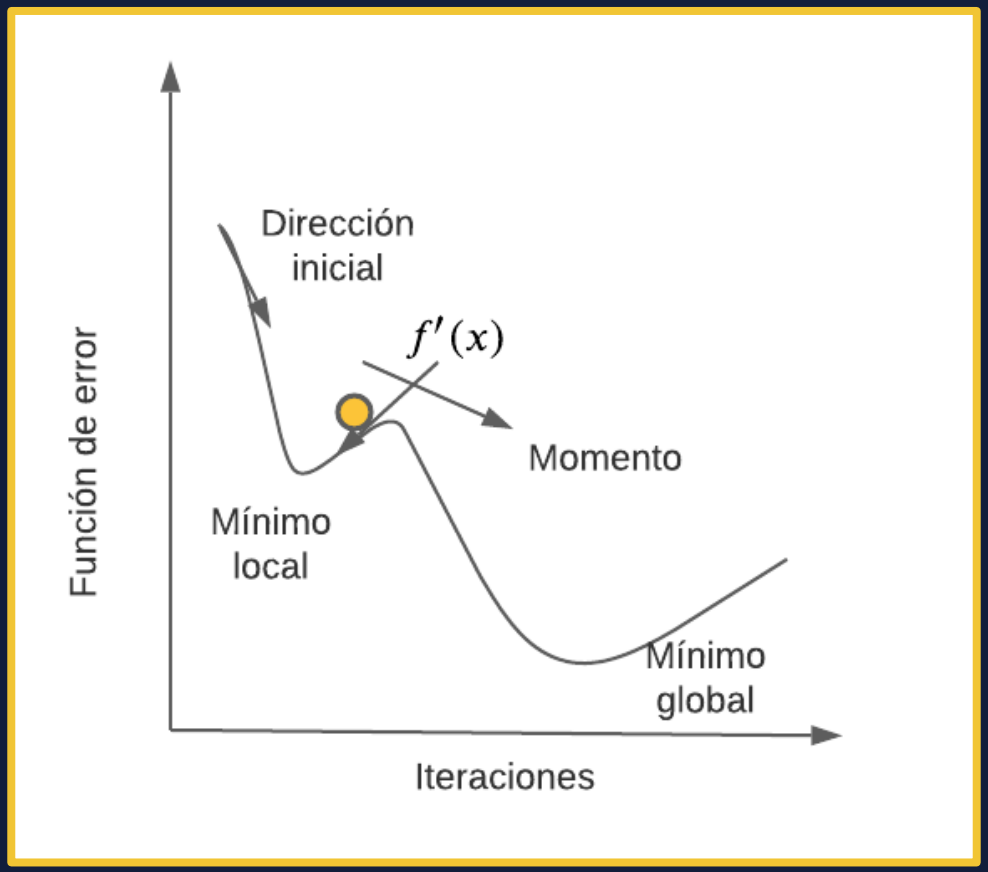
5. Dónde η y μ son la tasa de aprendizaje y el momento, respectivamente.
6. Repetir el procedimiento hasta que el algoritmo converja.

2. Redes neuronales artificiales – Entrenamiento

- **Tasa de aprendizaje (η):** controla que los pasos que se van dando no sean muy pequeños o muy grandes. Ajustar el valor de η es difícil, ya que si es demasiado grande podemos provocar oscilaciones, mientras que si es demasiado pequeño, necesitamos de muchas iteraciones.

- **Momento (μ):** El concepto de momento mejora la convergencia y permite que los cambios anteriores afecten en la dirección del movimiento actual. La idea es que el momento haga que se pueda escapar del óptimo local.

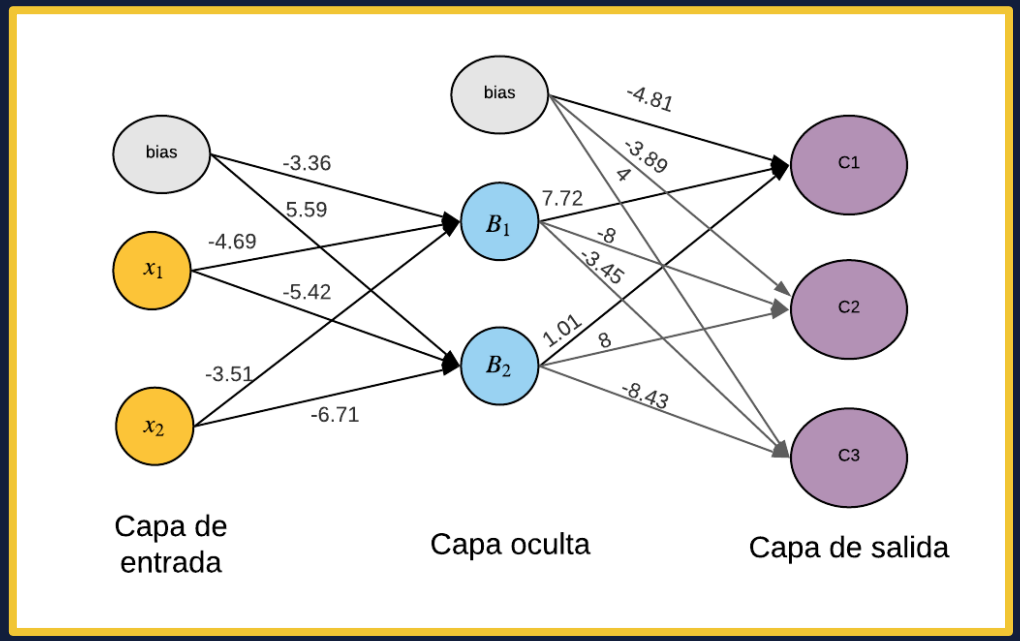
2. Redes neuronales artificiales – Entrenamiento



2. Redes neuronales artificiales – Consideraciones importantes

- Se trata de un modelo probabilístico no lineal.
- No obstante, la capa de salida se puede formular como una función lineal de funciones base no lineales.
- Se puede utilizar tanto en regresión como clasificación.
- Pueden sobreentrenar.
- El coste de entrenamiento es alto.
- Además, se necesitan estimar varios hiperparámetros: número de capas ocultas, número de neuronas en capa oculta, número de iteraciones, tasa de aprendizaje y momento, entre otros.
- Normalmente, son difíciles de interpretar.

2. Redes neuronales artificiales – Ejemplo



Patrón	X_1	X_2	Clase
1	1.4	0.2	1
2	1.5	0.2	1
3	1.5	0.4	1
4	1.6	0.6	2
5	4.0	1.3	2
6	4.2	1.2	2
7	4.1	1.3	3
8	5.1	1.9	3
9	5.0	1.9	3
10	5.1	1.8	3

1. Hallar las predicciones del modelo para el siguiente conjunto de *test.*
2. Evaluar el rendimiento del clasificador en dicho conjunto.

2. Redes neuronales artificiales – Ejemplo

1. Calculamos las predicciones.

Patrón	X_1	X_2	B_1	B_2	$p(C1)$	$p(C2)$	$p(C3)$	Predicha	Real
1	1.4	0.2	0.97	1.00	0.98	0.02	0.00	1	1
2	1.5	0.2	0.97	1.00	0.98	0.02	0.00	1	1
3	1.5	0.4	0.94	1.00	0.97	0.03	0.00	1	1
4	1.6	0.6	0.86	1.00	0.95	0.05	0.00	1	2
5	4.0	1.3	0.01	0.99	0.03	0.98	0.01	2	2
6	4.2	1.2	0.01	0.99	0.03	0.98	0.01	2	2
7	4.1	1.3	0.01	0.99	0.03	0.98	0.01	2	3
8	5.1	1.9	0.00	0.18	0.01	0.09	0.92	3	3
9	5.0	1.9	0.00	0.20	0.01	0.10	0.91	3	3
10	5.1	1.8	0.00	0.32	0.01	0.22	0.78	3	3

2. Redes neuronales artificiales – Ejemplo

2. Calculamos la matriz de confusión general, y para cada una de las clases. Después, obtenemos las métricas de rendimiento.

$$\begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix}$$

$$CCR = \frac{1}{n} \sum_{j=1}^{J} n_{jj} = 80\%$$

$$Kappa = \frac{p_0 - p_e}{1 - p_e} = \frac{CCR - \frac{1}{n^2} \sum_{j=1}^{J} n_{j\cdot}\, n_{\cdot j}}{1 - \frac{1}{n^2} \sum_{j=1}^{J} n_{j\cdot}\, n_{\cdot j}} = \frac{0.8 - 0.33}{1 - 0.33} = 0.70$$

2. Redes neuronales artificiales – Ejemplo

2. Calculamos la matriz de confusión general, y para cada una de las clases. Después, obtenemos las métricas.

Clase 1: $\begin{pmatrix} 3 & 0 \\ 1 & 6 \end{pmatrix}$ Clase 2: $\begin{pmatrix} 2 & 1 \\ 1 & 6 \end{pmatrix}$ Clase 3: $\begin{pmatrix} 3 & 1 \\ 0 & 6 \end{pmatrix}$

Clase	$\boldsymbol{Sensibilidad}$	$\boldsymbol{FP\ Rate}$	$\boldsymbol{Especifidad}$	$\boldsymbol{Precision}$	$\boldsymbol{F-Score}$
1	1	0.143	0.857	0.750	0.857
2	0.667	0.143	0.857	0.667	0.667
3	0.750	0	1	1	0.857
Promedio	**0.806**	**0.095**	**0.905**	**0.806**	**0.794**

3. Máquinas de vectores soporte – Fundamentos y modelo

- Un modelo de SVM es una representación de ejemplos como puntos en el espacio, mapeados de forma que dichos ejemplos pertenecientes a distintas categorías estén divididos por un claro espacio o margen que debe ser lo más ancho posible. Los nuevos patrones son después mapeados en el mismo espacio y la predicción se basa en determinar en qué lado del margen caen.

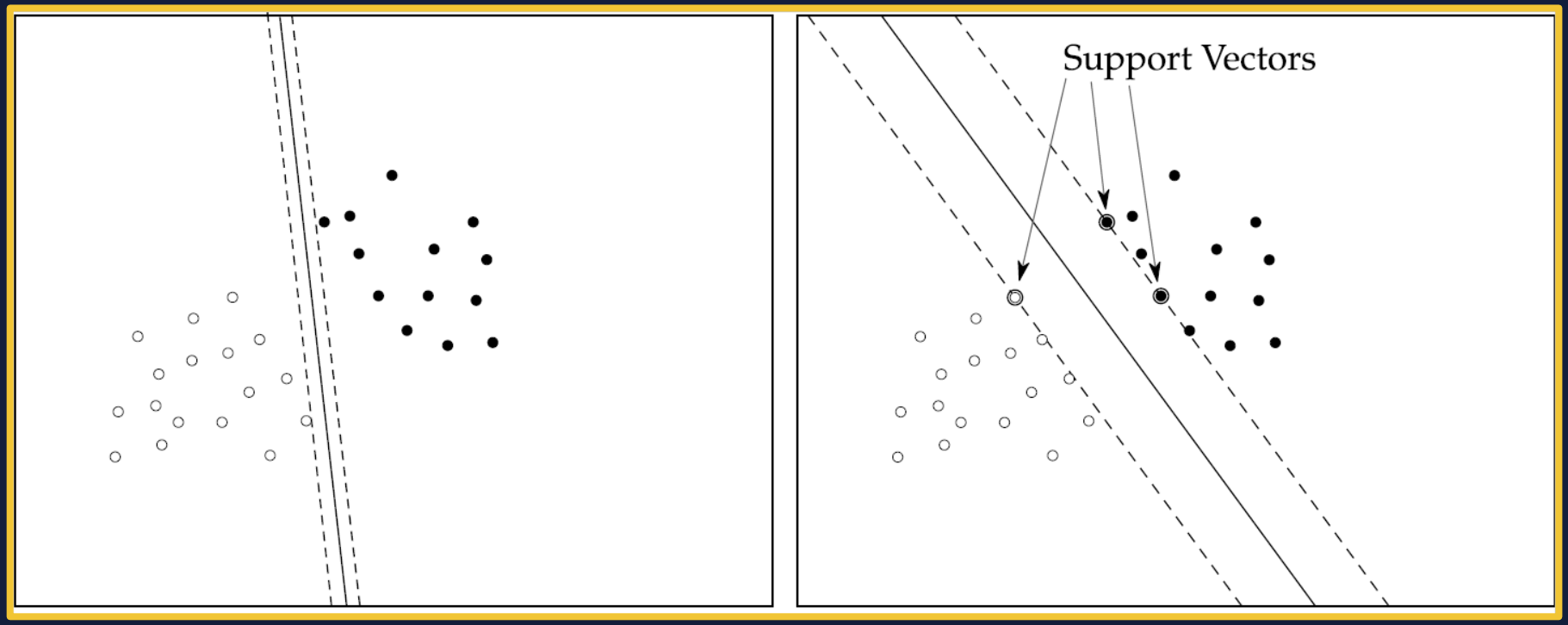
- La idea básica de las máquinas de vector soporte es encontrar el hiperplano que define la máxima separación entre patrones de dos clases diferentes. Formalmente:

$$f(x) = \hat{y} = sgn(< w \cdot \phi(x) > + b)$$

donde $\hat{y} = +1$ si x corresponde a la clase positiva y $\hat{y} = -1$ en otro caso.

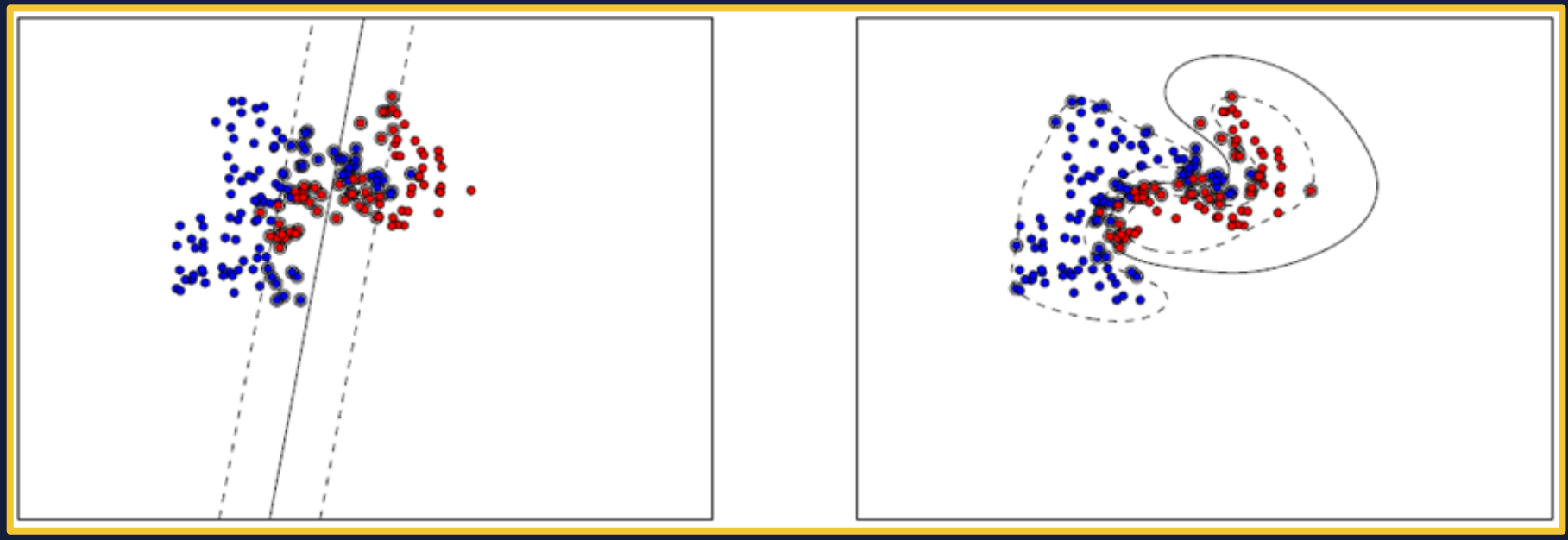
3. Máquinas de vectores soporte – Fundamentos y modelo

Caso lineal



3. Máquinas de vectores soporte – Fundamentos y modelo

Caso no lineal



3. Máquinas de vectores soporte – Entrenamiento

En resumen:

- SVM intenta resolver un problema de optimización intentando aumentar la distancia del margen de decisión entre las clases y maximizando el número de puntos correctamente clasificados.

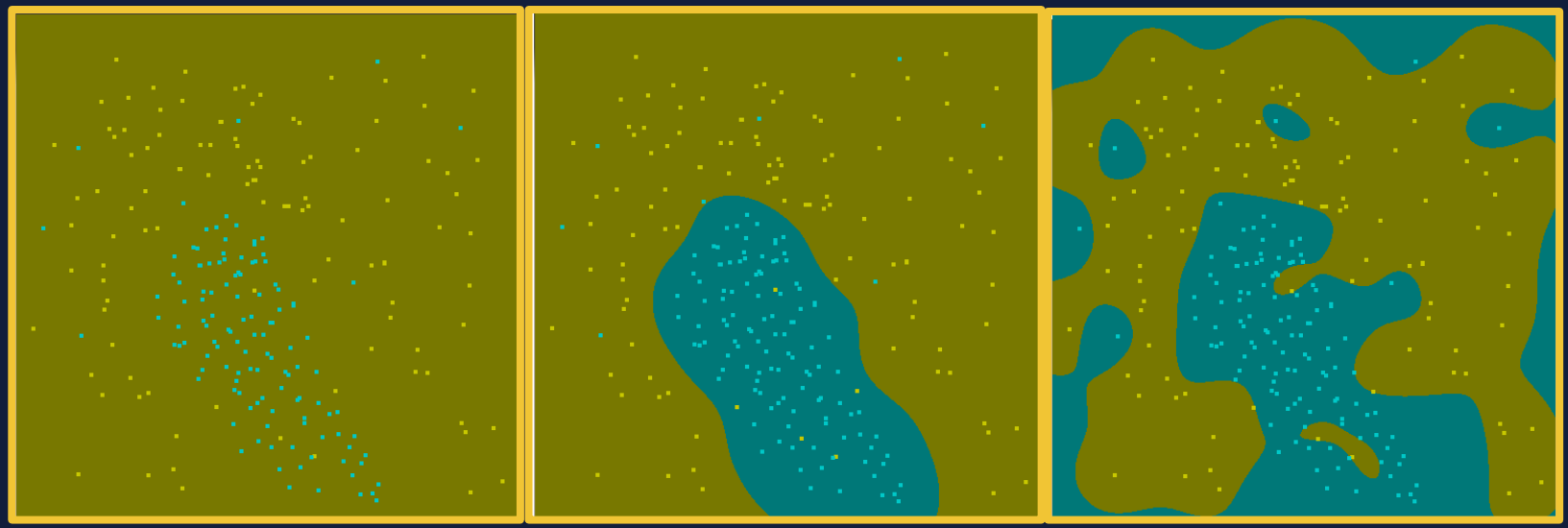
- El modelo de cálculo del hiperplano para el caso no lineal es complejo y requiere de habilidades matemáticas de optimización.

- Por tanto, en esta asignatura nos centraremos en la idea que subyace y fundamenta a los SVM, así como los hiperparámetros que son necesarios ajustar y que efecto provocan en nuestro clasificador.

3. Máquinas de vectores soporte – Entrenamiento

El primer hiperparámetro es el coste c:

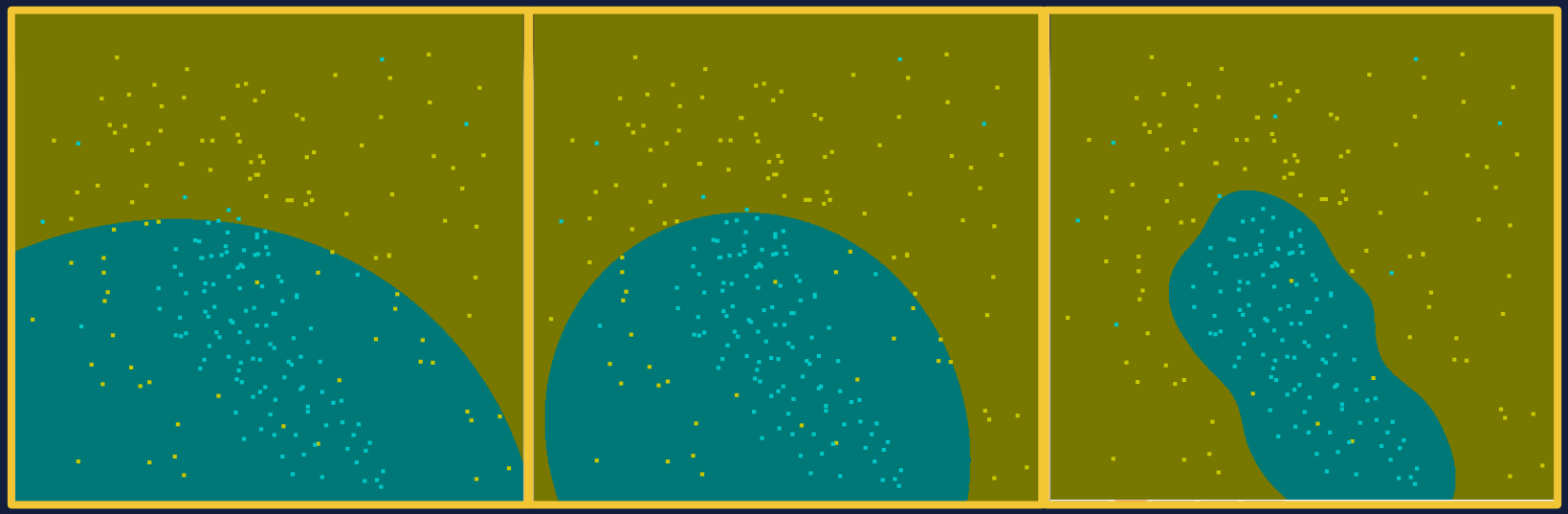
- Si es pequeño, la penalización por puntos mal clasificados es baja. Esto supondría un margen más ancho a pesar de un mayor número de errores de clasificación.
- Si es grande, el clasificador trataría de reducir los errores de clasificación con un margen más pequeño (valores altos pueden sobre entrenar).



3. Máquinas de vectores soporte – Entrenamiento

El segundo hiperparámetro es gamma γ:

- Presente en SVM no lineales con funciones RBF.
- Controla la distancia de la influencia de un solo punto de entrenamiento.
- Valores bajos implican un radio grande de similitud por lo que se agrupan más puntos.
- Valores altos implican que los puntos deben estar muy cerca para ser considerados del mismo grupo (valores altos pueden sobreentrenar).

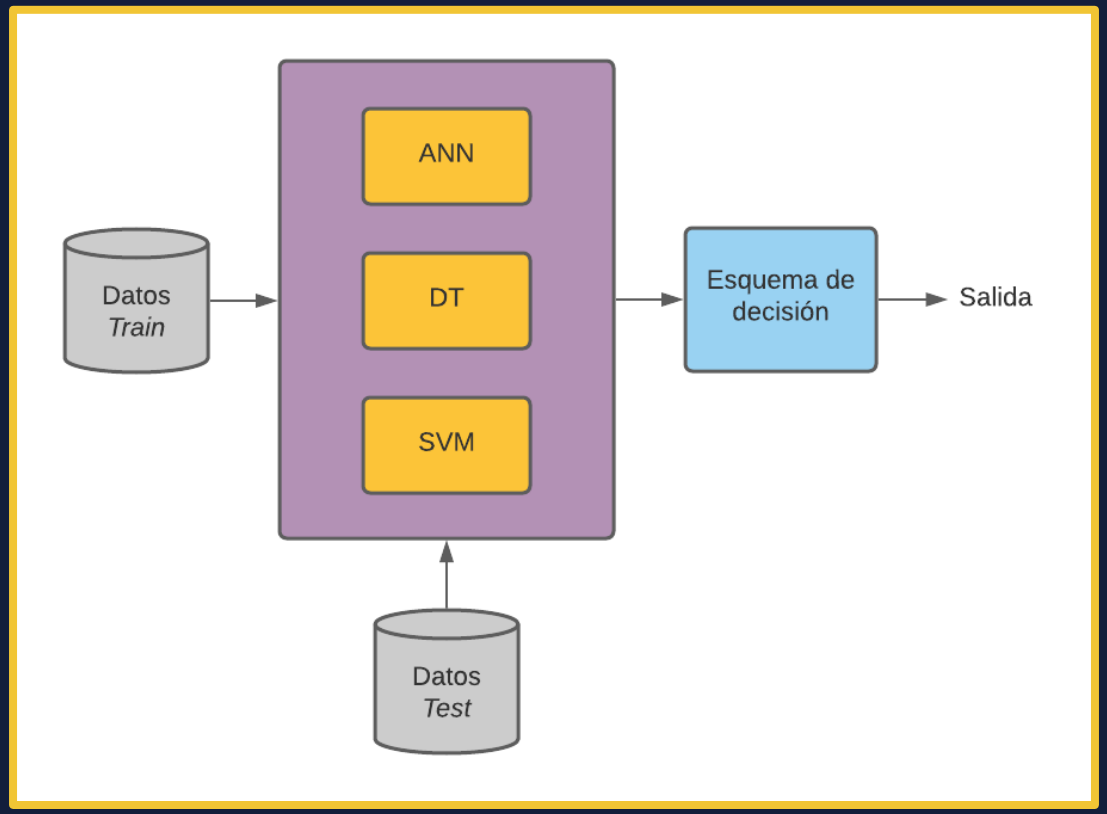


3. Máquinas de vectores soporte – Consideraciones importantes

- SVM trata de encontrar el hiperplano que mejor separa dos clases.
- Se pueden utilizar en clasificación (SVM) o regresión (SVR).
- En general, su rendimiento es muy bueno para bases de datos con pocas instancias y muchas características.
- Se dice que para pocos patrones (< 2000 – 10000 dependiendo de la dimensionalidad) los modelos no lineales son muy potentes, previo ajuste de los hiperparámetros c y γ.
- Para bases de datos mayores, el modelo lineal es bastante competitivo.
- El modelo es no lineal y es más robusto al problema de alta dimensionalidad.
- Hay que ajustar bien los parámetros para evitar el problema del sobreentrenamiento.
- No son interpretables.
- Aunque existen propuestas para la resolución distribuida del problema de optimización, suelen ser ineficientes para grandes volúmenes de datos.

4. Ensembles – Fundamentos y modelo

Se define un ensemble como un conjunto de modelos diferentes entre sí que unen sus predicciones con el propósito de hacer una predicción más robusta.



4. Ensembles – Fundamentos y modelo

- Se basan en la premisa de que cuantos más elementos decidan más robusta será la decisión.

- Los clasificadores o regresores suelen entrenar con procedimientos que pueden caer en óptimos locales. La agregación de sus predicciones podría acercarnos al óptimo global.

- Si distinguimos entre modelo fuerte como aquél que tiene una precisión deseada y modelo débil como aquél ligeramente superior a un modelo aleatorio, la idea es hacer ensembles de modelos débiles para construir un modelo fuerte.

4. Ensembles – Diversidad

Se considera que un ensemble aporta más que un modelo único en base a la **diversidad** de modelos. La intuición dice que los predictores base deben ser precisos y no cometer errores coincidentes.

- **Utilizando diferentes modelos** (árboles, ANN, SVM, …). Por lo general, a mayor número de predictores base, mejor será la predicción del ensemble, sin embargo, esto supone un gran coste computacional.

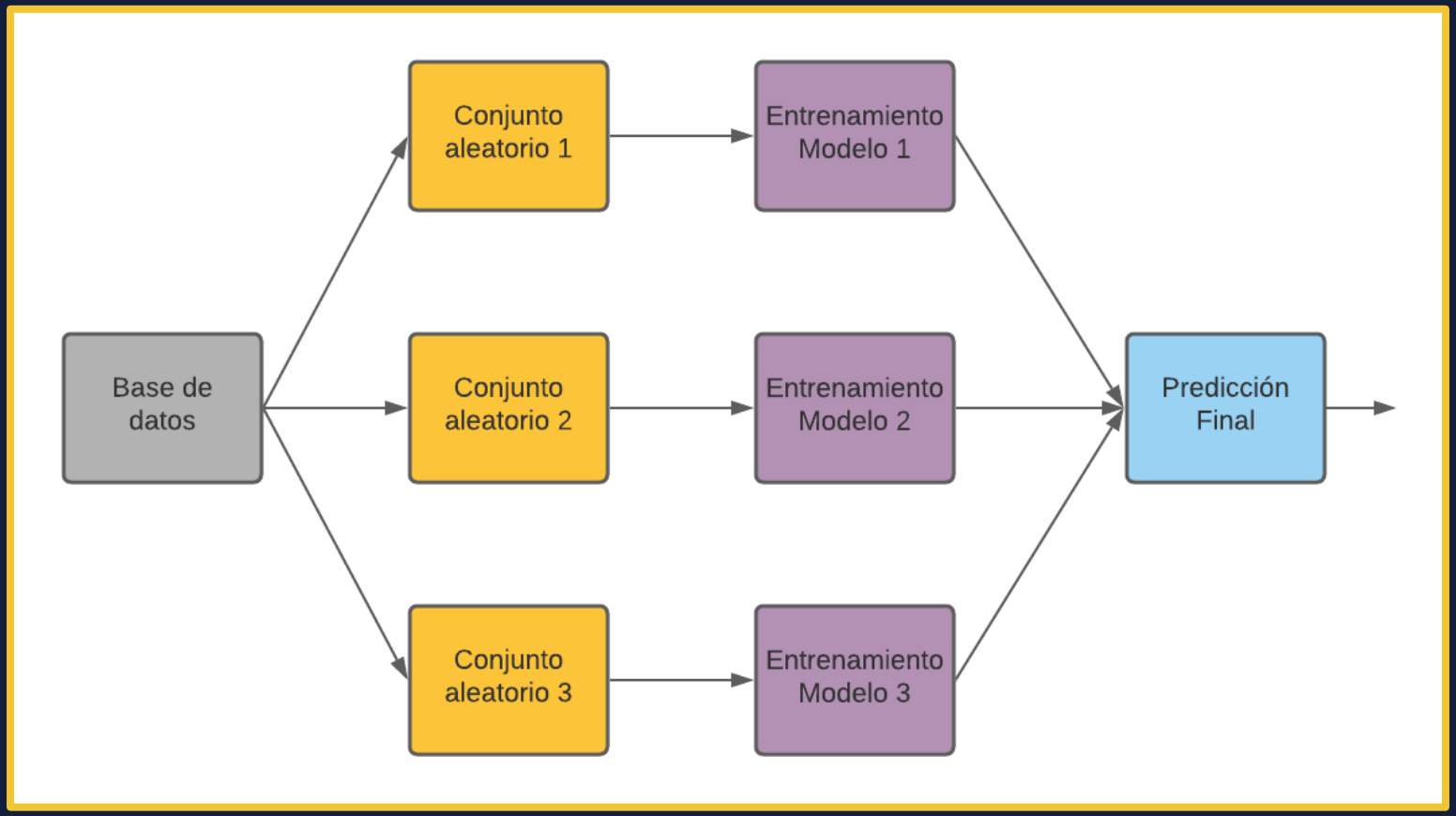
- **Manipulando los predictores**. Por ejemplo, teniendo configuraciones iniciales distintas de los predictores o manipulando los parámetros de entrenamiento.

- **Diferencia en la población de los datos**. Seleccionar un conjunto de datos distinto a cada predictor base.

- **Selección de características**. Los diferentes predictores pueden entrenarse sobre diferentes atributos de los patrones.

4. Ensembles – Generación de ensembles

Bagging



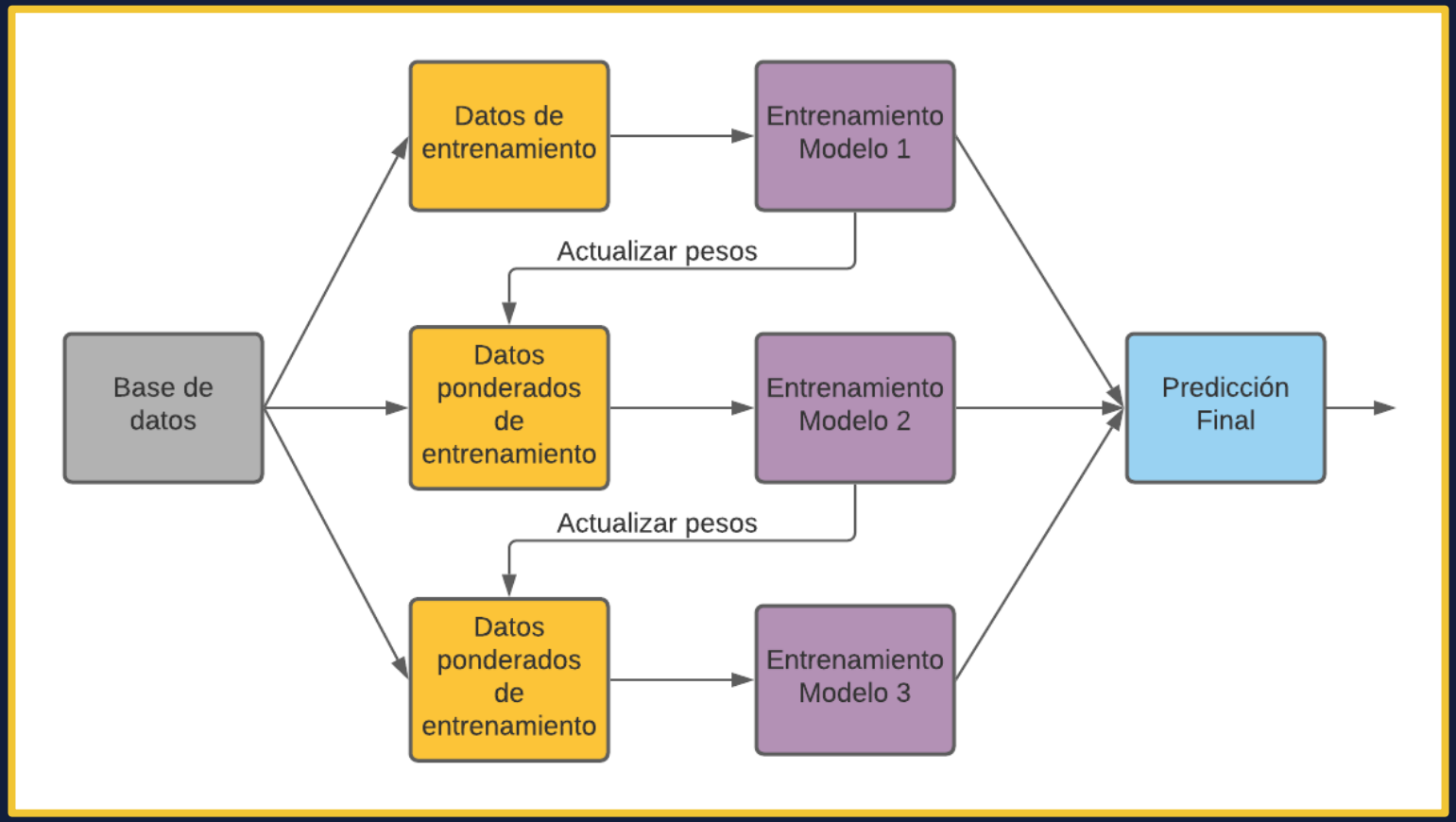
4. Ensembles – Generación de ensembles

Boosting - Adaboost

1. Entrenar un predictor (ajustar sus parámetros).
2. Usar el predictor para identificar los casos incorrectamente predichos.
3. Construir un nuevo predictor que mejore la predicción en los casos incorrectamente predichos del punto 2.
4. Repita los pasos 2 y 3 hasta que se satisface la condición de parada (número de clasificadores base utilizados o patrones correctamente predichos).
5. Asignar un peso a cada predictor y unirlos para obtener un modelo con mejor desempeño.

4. Ensembles – Generación de ensembles

Boosting - Adaboost



4. Ensembles – Consideraciones importantes

- Los ensembles unen las predicciones de un conjunto de modelos base.
- Funcionan tanto en regresión como clasificación.
- Debe haber diversidad en cada modelo base.
- Dos de las técnicas de generación de ensembles más utilizadas son *Bagging* y *Boosting*.
- Tanto el tiempo de entrenamiento como el tiempo de test son obviamente mayores que en modelos únicos. Dependerá de los requisitos de nuestro problema elegir este tipo de técnicas.

MUCHAS GRACIAS POR SU ATENCIÓN

eip ESCUELA INTERNACIONAL POSGRADOS

aduran@grupomainjobs.com

Antonio Manuel Durán Rosal
https://www.linkedin.com/in/antonio-manuel-dur%C3%A1n-rosal-99ba92a5/

twitter.com/eiposgrados

facebook.com/eiposgrados

instagram.com/eiposgrados